Foundation of Cryptography (0368-4162-01), Lecture 4

Interactive Proofs and Zero Knowledge

Iftach Haitner, Tel Aviv University

November 29, 2011

Part I

Interactive Proofs

Interactive Vs. Interactive Proofs

Definition 1 (NP)

$\mathcal{L} \in \mathrm{NP}$ iff $\exists \ell \in$ poly and poly-time algorithm V such that:

- $\forall x \in \mathcal{L} \cap \{0,1\}^n$ there exists $w \in \{0,1\}^{\ell(n)}$ s.t. $\mathsf{V}(x, w) = 1$
- $\mathsf{V}(x, \cdot) = 0$ for every $x \notin \mathcal{L}$

Interactive Vs. Interactive Proofs

Definition 1 (NP)

$\mathcal{L} \in \mathrm{NP}$ iff $\exists \ell \in$ poly and poly-time algorithm V such that:

- $\forall x \in \mathcal{L} \cap \{0,1\}^n$ there exists $w \in \{0,1\}^{\ell(n)}$ s.t. $\mathsf{V}(x,w) = 1$
- $\mathsf{V}(x,\cdot) = 0$ for every $x \notin \mathcal{L}$

- *Non-interactive* proof

Interactive Vs. Interactive Proofs

Definition 1 (NP)

$\mathcal{L} \in \mathrm{NP}$ iff $\exists \ell \in$ poly and poly-time algorithm V such that:

- $\forall x \in \mathcal{L} \cap \{0,1\}^n$ there exists $w \in \{0,1\}^{\ell(n)}$ s.t. $\mathrm{V}(x,w) = 1$
- $\mathrm{V}(x,\cdot) = 0$ for every $x \notin \mathcal{L}$

- *Non-interactive* proof
- Interactive proofs?

Interactive Vs. Interactive Proofs

Definition 1 (NP)

$\mathcal{L} \in \mathrm{NP}$ iff $\exists \ell \in$ poly and poly-time algorithm V such that:

- $\forall x \in \mathcal{L} \cap \{0,1\}^n$ there exists $w \in \{0,1\}^{\ell(n)}$ s.t. $\mathsf{V}(x,w) = 1$
- $\mathsf{V}(x,\cdot) = 0$ for every $x \notin \mathcal{L}$

- *Non-interactive* proof
- Interactive proofs?

Interactive protocols

- Interactive algorithm

Interactive protocols

- Interactive algorithm
- Protocol $\pi = (\mathsf{A}, \mathsf{B})$

Interactive protocols

- Interactive algorithm
- Protocol $\pi = (\mathsf{A}, \mathsf{B})$
- RV describing the parties joint output $\langle \mathsf{A}(i_\mathsf{A}), \mathsf{B}(i_\mathsf{B}))(i) \rangle$

Interactive protocols

- Interactive algorithm
- Protocol $\pi = (\mathsf{A}, \mathsf{B})$
- RV describing the parties joint output $\langle \mathsf{A}(i_\mathsf{A}), \mathsf{B}(i_\mathsf{B}))(i) \rangle$
- *m*-round algorithm, *m*-round protocol

Interactive Proofs

Definition 2 (Interactive Proof (IP))

A protocol (P, V) is an interactive proof for $\mathcal{L}$, if V is $\mathtt{PPT}$ and the following hold:

Completeness $\forall x \in \mathcal{L}$, $\Pr[\langle(\mathsf{P}, \mathsf{V})(x)\rangle = \mathtt{Accept}] \geq 2/3$

Soundness $\forall x \notin \mathcal{L}$, and *any* algorithm P^*
$\Pr[\langle(\mathsf{P}^*, \mathsf{V})(x)\rangle = \mathtt{Accept}] \leq 1/3$

Interactive Proofs

Definition 2 (Interactive Proof (IP))

A protocol (P, V) is an interactive proof for $\mathcal{L}$, if V is $\mathtt{PPT}$ and the following hold:

Completeness $\forall x \in \mathcal{L}$, $\Pr[\langle(\mathsf{P}, \mathsf{V})(x)\rangle = \mathtt{Accept}] \geq 2/3$

Soundness $\forall x \notin \mathcal{L}$, and *any* algorithm P^*
$\Pr[\langle(\mathsf{P}^*, \mathsf{V})(x)\rangle = \mathtt{Accept}] \leq 1/3$

- $\mathrm{IP} = \mathrm{PSPACE}$

Interactive Proofs

Definition 2 (Interactive Proof (IP))

A protocol (P, V) is an interactive proof for $\mathcal{L}$, if V is $\texttt{PPT}$ and the following hold:

Completeness $\forall x \in \mathcal{L}$, $\Pr[\langle(\mathsf{P}, \mathsf{V})(x)\rangle = \texttt{Accept}] \geq 2/3$

Soundness $\forall x \notin \mathcal{L}$, and *any* algorithm P^*
$\Pr[\langle(\mathsf{P}^*, \mathsf{V})(x)\rangle = \texttt{Accept}] \leq 1/3$

- $\mathrm{IP} = \mathrm{PSPACE}$
- We typically consider (and achieve) perfect completeness

Interactive Proofs

Definition 2 (Interactive Proof (IP))

A protocol (P, V) is an interactive proof for $\mathcal{L}$, if V is $\texttt{PPT}$ and the following hold:

Completeness $\forall x \in \mathcal{L}$, $\Pr[\langle(\mathsf{P}, \mathsf{V})(x)\rangle = \texttt{Accept}] \geq 2/3$

Soundness $\forall x \notin \mathcal{L}$, and *any* algorithm P^*
$\Pr[\langle(\mathsf{P}^*, \mathsf{V})(x)\rangle = \texttt{Accept}] \leq 1/3$

- $\mathrm{IP} = \mathrm{PSPACE}$
- We typically consider (and achieve) perfect completeness
- Negligible “soundness error" achieved via repetition.

Interactive Proofs

Definition 2 (Interactive Proof (IP))

A protocol (P, V) is an interactive proof for $\mathcal{L}$, if V is $\texttt{PPT}$ and the following hold:

Completeness $\forall x \in \mathcal{L}$, $\Pr[\langle(\mathsf{P}, \mathsf{V})(x)\rangle = \texttt{Accept}] \geq 2/3$

Soundness $\forall x \notin \mathcal{L}$, and *any* algorithm P^*
$\Pr[\langle(\mathsf{P}^*, \mathsf{V})(x)\rangle = \texttt{Accept}] \leq 1/3$

- $\mathrm{IP} = \mathrm{PSPACE}$
- We typically consider (and achieve) perfect completeness
- Negligible “soundness error" achieved via repetition.
- soundness only against $\texttt{PPT}$: *computationally sound proofs/interactive arguments*.

Interactive Proofs

Definition 2 (Interactive Proof (IP))

A protocol (P, V) is an interactive proof for $\mathcal{L}$, if V is $\mathtt{PPT}$ and the following hold:

Completeness $\forall x \in \mathcal{L}$, $\Pr[\langle(\mathsf{P}, \mathsf{V})(x)\rangle = \mathtt{Accept}] \geq 2/3$

Soundness $\forall x \notin \mathcal{L}$, and *any* algorithm P^*
$\Pr[\langle(\mathsf{P}^*, \mathsf{V})(x)\rangle = \mathtt{Accept}] \leq 1/3$

- $\mathrm{IP} = \mathrm{PSPACE}$
- We typically consider (and achieve) perfect completeness
- Negligible "soundness error" achieved via repetition.
- soundness only against $\mathtt{PPT}$: *computationally sound proofs/interactive arguments*.
- efficient provers via "auxiliary input"

Section 1

IP for GNI

graph isomorphism

Π_m – the set of all permutations from $[m]$ to $[m]$

Definition 3 (graph isomorphism)

Graphs $G_0 = ([m], E_0)$ and $G_1 = ([m], E_1)$ are *isomorphic*, denoted $G_0 \equiv G_1$, if $\exists \pi \in \Pi_m$ such that $(u, v) \in E_0$ iff $(\pi(u), \pi(v)) \in E_1$.
$\mathrm{GI} = \{(G_0, G_1) \colon G_0 \equiv G_1\}$.

graph isomorphism

Π_m – the set of all permutations from $[m]$ to $[m]$

Definition 3 (graph isomorphism)

Graphs $G_0 = ([m], E_0)$ and $G_1 = ([m], E_1)$ are *isomorphic*, denoted $G_0 \equiv G_1$, if $\exists \pi \in \Pi_m$ such that $(u, v) \in E_0$ iff $(\pi(u), \pi(v)) \in E_1$.
$\mathsf{GI} = \{(G_0, G_1) \colon G_0 \equiv G_1\}$.

- Assume reasonable mapping from graphs to strings

graph isomorphism

Π_m – the set of all permutations from $[m]$ to $[m]$

Definition 3 (graph isomorphism)

Graphs $G_0 = ([m], E_0)$ and $G_1 = ([m], E_1)$ are *isomorphic*, denoted $G_0 \equiv G_1$, if $\exists \pi \in \Pi_m$ such that $(u, v) \in E_0$ iff $(\pi(u), \pi(v)) \in E_1$.
$\mathsf{GI} = \{(G_0, G_1) \colon G_0 \equiv G_1\}$.

- Assume reasonable mapping from graphs to strings
- $\mathsf{GI} \in \mathrm{NP}$

graph isomorphism

Π_m – the set of all permutations from $[m]$ to $[m]$

Definition 3 (graph isomorphism)

Graphs $G_0 = ([m], E_0)$ and $G_1 = ([m], E_1)$ are *isomorphic*, denoted $G_0 \equiv G_1$, if $\exists \pi \in \Pi_m$ such that $(u, v) \in E_0$ iff $(\pi(u), \pi(v)) \in E_1$.
$\mathrm{GI} = \{(G_0, G_1) \colon G_0 \equiv G_1\}$.

- Assume reasonable mapping from graphs to strings
- $\mathrm{GI} \in \mathrm{NP}$
- Does $\mathrm{GNI} = \{(G_0, G_1) \colon G_0 \not\equiv G_1\} \in \mathrm{NP}$?

graph isomorphism

Π_m – the set of all permutations from $[m]$ to $[m]$

Definition 3 (graph isomorphism)

Graphs $G_0 = ([m], E_0)$ and $G_1 = ([m], E_1)$ are *isomorphic*, denoted $G_0 \equiv G_1$, if $\exists \pi \in \Pi_m$ such that $(u, v) \in E_0$ iff $(\pi(u), \pi(v)) \in E_1$.
$\mathrm{GI} = \{(G_0, G_1) \colon G_0 \equiv G_1\}$.

- Assume reasonable mapping from graphs to strings
- $\mathrm{GI} \in \mathrm{NP}$
- Does $\mathrm{GNI} = \{(G_0, G_1) \colon G_0 \not\equiv G_1\} \in \mathrm{NP}$?
- We will show a simple interactive proof for GNI

graph isomorphism

Π_m – the set of all permutations from $[m]$ to $[m]$

Definition 3 (graph isomorphism)

Graphs $G_0 = ([m], E_0)$ and $G_1 = ([m], E_1)$ are *isomorphic*, denoted $G_0 \equiv G_1$, if $\exists \pi \in \Pi_m$ such that $(u, v) \in E_0$ iff $(\pi(u), \pi(v)) \in E_1$.
$\mathrm{GI} = \{(G_0, G_1) \colon G_0 \equiv G_1\}$.

- Assume reasonable mapping from graphs to strings
- $\mathrm{GI} \in \mathrm{NP}$
- Does $\mathrm{GNI} = \{(G_0, G_1) \colon G_0 \not\equiv G_1\} \in \mathrm{NP}$?
- We will show a simple interactive proof for GNI Idea: Beer tasting...

$\mathbb{IP}$ for GNI

Protocol 4 ((P, V))

Common input $G_0 = ([m], E_0), G_1 = ([m], E_1)$

1. V chooses $b \leftarrow \{0, 1\}$ and $\pi \leftarrow \Pi_m$, and sends $\pi(E_b) = \{(\pi(u), \pi(v)) \colon (u, v) \in E_b\}$ to P
2. P send b' to V (tries to set $b' = b$)
3. V accepts iff $b' = b$

IP for GNI

Protocol 4 ((P, V))

Common input $G_0 = ([m], E_0), G_1 = ([m], E_1)$

1. V chooses $b \leftarrow \{0,1\}$ and $\pi \leftarrow \Pi_m$, and sends $\pi(E_b) = \{(\pi(u), \pi(v)) \colon (u,v) \in E_b\}$ to P
2. P send b' to V (tries to set $b' = b$)
3. V accepts iff $b' = b$

Claim 5

The above protocol is IP for GNI, with perfect completeness and soundness error $\frac{1}{2}$.

Proving Claim 5

- Graph isomorphism is an equivalence relation (separates the set of all graph pairs into separate subsets)

Proving Claim 5

- Graph isomorphism is an equivalence relation (separates the set of all graph pairs into separate subsets)
- $([m], \pi(E_i))$ is a random element in $[G_i]$ — the equivalence class of G_i

Proving Claim 5

- Graph isomorphism is an equivalence relation (separates the set of all graph pairs into separate subsets)
- $([m], \pi(E_i))$ is a random element in $[G_i]$ — the equivalence class of G_i

Hence,

$G_0 \equiv G_1$: $\Pr[b' = b] \leq \frac{1}{2}$.

Proving Claim 5

- Graph isomorphism is an equivalence relation (separates the set of all graph pairs into separate subsets)
- $([m], \pi(E_i))$ is a random element in $[G_i]$ — the equivalence class of G_i

Hence,

$G_0 \equiv G_1$: $\Pr[b' = b] \le \frac{1}{2}$.

$G_0 \not\equiv G_1$: $\Pr[b' = b] = 1$ (i.e., i can, possibly inefficiently, extracted from $\pi(E_i)$)

□

Part II

Zero knowledge Proofs

The concept of zero knowledge

- Proving w/o revealing any addition information.

The concept of zero knowledge

- Proving w/o revealing any addition information.
- What does it mean?

The concept of zero knowledge

- Proving w/o revealing any addition information.
- What does it mean?
  Simulation paradigm.

Zero knowledge Proof

Definition 6 (computational ZK)

An interactive proof (P, V) is computational zero-knowledge proof (CZKP) for $\mathcal{L}$, if $\forall$ PPT V^*, $\exists$ PPT S such that $\{\langle(P, V^*)(x)\rangle\}_{x\in\mathcal{L}} \approx_c \{S(x)\}_{x\in\mathcal{L}}$.

Zero knowledge Proof

Definition 6 (computational ZK)

An interactive proof (P, V) is computational zero-knowledge proof (CZKP) for $\mathcal{L}$, if $\forall$ PPT V^*, $\exists$ PPT S such that $\{\langle(P, V^*)(x)\rangle\}_{x\in\mathcal{L}} \approx_c \{S(x)\}_{x\in\mathcal{L}}$.
Perfect ZK (PZKP)/statistical ZK (SZKP) – the above dist. are identicallly/statistically close, even for *unbounded* V^*.

Zero knowledge Proof

Definition 6 (computational ZK)

An interactive proof (P, V) is computational zero-knowledge proof (CZKP) for $\mathcal{L}$, if $\forall$ PPT V^*, $\exists$ PPT S such that $\{\langle(\mathsf{P}, \mathsf{V}^*)(x)\rangle\}_{x\in\mathcal{L}} \approx_c \{S(x)\}_{x\in\mathcal{L}}$.
Perfect ZK (PZKP)/statistical ZK (SZKP) – the above dist. are identicallly/statistically close, even for *unbounded* V^*.

1. ZK is a property of the prover.

Zero knowledge Proof

Definition 6 (computational ZK)

An interactive proof (P, V) is computational zero-knowledge proof (CZKP) for $\mathcal{L}$, if $\forall$ PPT V^*, $\exists$ PPT S such that $\{\langle(\mathsf{P}, \mathsf{V}^*)(x)\rangle\}_{x\in\mathcal{L}} \approx_c \{S(x)\}_{x\in\mathcal{L}}$.
Perfect ZK (PZKP)/statistical ZK (SZKP) – the above dist. are identicallly/statistically close, even for *unbounded* V^*.

1. ZK is a property of the prover.
2. ZK only required to hold with respect to true statements.

Zero knowledge Proof

Definition 6 (computational ZK)

An interactive proof (P, V) is computational zero-knowledge proof (CZKP) for $\mathcal{L}$, if $\forall$ PPT V^*, $\exists$ PPT S such that $\{\langle(\mathsf{P}, \mathsf{V}^*)(x)\rangle\}_{x\in\mathcal{L}} \approx_c \{S(x)\}_{x\in\mathcal{L}}$.
Perfect ZK (PZKP)/statistical ZK (SZKP) – the above dist. are identicallly/statistically close, even for *unbounded* V^*.

1. ZK is a property of the prover.
2. ZK only required to hold with respect to true statements.
3. wlg. V^*'s outputs is its “view".

Zero knowledge Proof

Definition 6 (computational ZK)

An interactive proof (P, V) is computational zero-knowledge proof (CZKP) for $\mathcal{L}$, if $\forall$ PPT V^*, $\exists$ PPT S such that $\{\langle(\mathsf{P}, \mathsf{V}^*)(x)\rangle\}_{x\in\mathcal{L}} \approx_c \{S(x)\}_{x\in\mathcal{L}}$.
Perfect ZK (PZKP)/statistical ZK (SZKP) – the above dist. are identicallly/statistically close, even for *unbounded* V^*.

1. ZK is a property of the prover.
2. ZK only required to hold with respect to true statements.
3. wlg. V^*'s outputs is its “view".
4. Trivial to achieve for $\mathcal{L} \in \mathrm{BPP}$

Zero knowledge Proof

Definition 6 (computational ZK)

An interactive proof (P, V) is computational zero-knowledge proof (CZKP) for $\mathcal{L}$, if $\forall$ PPT V^*, $\exists$ PPT S such that $\{\langle(\mathsf{P}, \mathsf{V}^*)(x)\rangle\}_{x\in\mathcal{L}} \approx_c \{S(x)\}_{x\in\mathcal{L}}$.
Perfect ZK (PZKP)/statistical ZK (SZKP) – the above dist. are identicallly/statistically close, even for *unbounded* V^*.

1. ZK is a property of the prover.
2. ZK only required to hold with respect to true statements.
3. wlg. V^*'s outputs is its “view".
4. Trivial to achieve for $\mathcal{L} \in \mathrm{BPP}$
5. Extension: P gets auxiliary input (e.g., NP witness)

Zero knowledge Proof

Definition 6 (computational ZK)

An interactive proof (P, V) is computational zero-knowledge proof (CZKP) for $\mathcal{L}$, if $\forall$ PPT V^*, $\exists$ PPT S such that $\{\langle(\mathsf{P}, \mathsf{V}^*)(x)\rangle\}_{x\in\mathcal{L}} \approx_c \{S(x)\}_{x\in\mathcal{L}}$.
Perfect ZK (PZKP)/statistical ZK (SZKP) – the above dist. are identicallly/statistically close, even for *unbounded* V^*.

1. ZK is a property of the prover.
2. ZK only required to hold with respect to true statements.
3. wlg. V^*'s outputs is its “view".
4. Trivial to achieve for $\mathcal{L} \in \mathrm{BPP}$
5. Extension: P gets auxiliary input (e.g., NP witness)
6. The “standard" NP proof is typically not zero knowledge

Zero knowledge Proof

Definition 6 (computational ZK)

An interactive proof (P, V) is computational zero-knowledge proof (CZKP) for $\mathcal{L}$, if $\forall$ PPT V^*, $\exists$ PPT S such that $\{\langle(\mathsf{P}, \mathsf{V}^*)(x)\rangle\}_{x \in \mathcal{L}} \approx_c \{S(x)\}_{x \in \mathcal{L}}$.
Perfect ZK (PZKP)/statistical ZK (SZKP) – the above dist. are identicallly/statistically close, even for *unbounded* V^*.

1. ZK is a property of the prover.
2. ZK only required to hold with respect to true statements.
3. wlg. V^*'s outputs is its “view".
4. Trivial to achieve for $\mathcal{L} \in \mathrm{BPP}$
5. Extension: P gets auxiliary input (e.g., NP witness)
6. The “standard" NP proof is typically not zero knowledge
7. Next class — ZK for all NP

Section 2

ZK Proof for GI

ZK Proof for Graph Isomorphism

Idea: route finding

ZK Proof for Graph Isomorphism

Idea: route finding

Protocol 7 ((P, V))

Common input $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$

P's input a permutation π such that $\pi(E_1) = E_0$

1. P chooses $\pi' \leftarrow \Pi_m$ and sends $E = \pi'(E_0)$ to V
2. V sends $b \leftarrow \{0, 1\}$ to P
3. if $b = 0$, P sets $\pi'' = \pi'$, otherwise, it sends $\pi'' = \pi' \circ \pi$ to V
4. V accepts iff $\pi''(E_b) = E$

ZK Proof for Graph Isomorphism

Idea: route finding

Protocol 7 ((P, V))

Common input $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$

P's input a permutation π such that $\pi(E_1) = E_0$

1. P chooses $\pi' \leftarrow \Pi_m$ and sends $E = \pi'(E_0)$ to V
2. V sends $b \leftarrow \{0, 1\}$ to P
3. if $b = 0$, P sets $\pi'' = \pi'$, otherwise, it sends $\pi'' = \pi' \circ \pi$ to V
4. V accepts iff $\pi''(E_b) = E$

Claim 8

The above protocol is $\mathcal{SZKP}$ for $\mathcal{GI}$, with perfect completeness and soundness $\frac{1}{2}$.

Proving Claim 8

Completeness Clear

Proving Claim 8

Completeness Clear

Soundness If exist $j \in \{0, 1\}$ for which $\nexists \pi' \in \Pi_m$ with $\pi'(E_j) = E$, then V rejects w.p. at least $\frac{1}{2}$.

Proving Claim 8

Completeness Clear

Soundness If exist $j \in \{0,1\}$ for which $\nexists \pi' \in \Pi_m$ with $\pi'(E_j) = E$, then V rejects w.p. at least $\frac{1}{2}$.
Assuming V rejects w.p. less than $\frac{1}{2}$ and lett π_0 and π_1 be the values guaranteed by the above observation (i.e., mapping E_0 and E_1 to E respectively).

Proving Claim 8

Completeness Clear

Soundness If exist $j \in \{0,1\}$ for which $\nexists \pi' \in \Pi_m$ with $\pi'(E_j) = E$, then V rejects w.p. at least $\frac{1}{2}$.
Assuming V rejects w.p. less than $\frac{1}{2}$ and lett π_0 and π_1 be the values guaranteed by the above observation (i.e., mapping E_0 and E_1 to E respectively).
Then $\pi_0^{-1}(\pi_1(E_1)) = \pi_0$

Proving Claim 8

Completeness Clear

Soundness If exist $j \in \{0,1\}$ for which $\nexists \pi' \in \Pi_m$ with $\pi'(E_j) = E$, then V rejects w.p. at least $\frac{1}{2}$.
Assuming V rejects w.p. less than $\frac{1}{2}$ and lett π_0 and π_1 be the values guaranteed by the above observation (i.e., mapping E_0 and E_1 to E respectively).
Then $\pi_0^{-1}(\pi_1(E_1)) = \pi_0 \implies (G_0, G_1) \in \mathrm{GI}$.

Proving Claim 8

Completeness Clear

Soundness If exist $j \in \{0,1\}$ for which $\nexists \pi' \in \Pi_m$ with $\pi'(E_j) = E$, then V rejects w.p. at least $\frac{1}{2}$.
Assuming V rejects w.p. less than $\frac{1}{2}$ and lett π_0 and π_1 be the values guaranteed by the above observation (i.e., mapping E_0 and E_1 to E respectively).
Then $\pi_0^{-1}(\pi_1(E_1)) = \pi_0 \implies (G_0, G_1) \in \mathrm{GI}$.

ZK Idea: for $(G_0, G_1) \in \mathrm{GI}$, it is easy to generate a random transcript for Steps 1-2, and to be able to open it with prob $\frac{1}{2}$.

The simulator

For a start we consider a deterministic cheating verifier V^* that never aborts.

The simulator

For a start we consider a deterministic cheating verifier V^* that never aborts.

Algorithm 9 (S)

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$
Do $|x|$ times:

1. Choose $b' \leftarrow \{0,1\}$ and $\pi \leftarrow \Pi_m$, and “send" $\pi(E_{b'})$ to $V^*(x)$.
2. Let b be V^*'s answer. If $b = b'$, send π to V^*, and output V^*'s output.
   Otherwise, rewind the simulation to its first step.

Abort

The simulator

For a start we consider a deterministic cheating verifier V^* that never aborts.

Algorithm 9 (S)

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$
Do $|x|$ times:

1. Choose $b' \leftarrow \{0,1\}$ and $\pi \leftarrow \Pi_m$, and "send" $\pi(E_{b'})$ to $\mathsf{V}^*(x)$.
2. Let b be V^*'s answer. If $b = b'$, send π to V^*, and output V^*'s output.
   Otherwise, rewind the simulation to its first step.

Abort

Claim 10

$\{\langle(\mathsf{P}, \mathsf{V}^*)(x)\rangle\}_{x \in \mathsf{GI}} \approx \{\mathsf{S}(x)\}_{x \in \mathsf{GI}}$

Proving Claim 10

Algorithm 11 (S')

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$
Do $|x|$ times:

1. Choose $\pi \leftarrow \Pi_m$ and sends $E = \pi(E_0)$ to $V^*(x)$.
2. Let b be V^*'s answer.
   w.p. $\frac{1}{2}$, find π' such that $E = \pi'(E_b)$, send it to V^* and output V^*'s output.
   Otherwise, rewind the simulation to its first step.

Abort

Proving Claim 10

Algorithm 11 (S')

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$
Do $|x|$ times:

1. Choose $\pi \leftarrow \Pi_m$ and sends $E = \pi(E_0)$ to $\mathsf{V}^*(x)$.
2. Let b be V^*'s answer.
   w.p. $\frac{1}{2}$, find π' such that $E = \pi'(E_b)$, send it to V^* and output V^*'s output.
   Otherwise, rewind the simulation to its first step.

Abort

Claim 12

$\mathsf{S}(x) \equiv \mathsf{S}'(x)$ for any $x \in \mathrm{GI}$.

Proving Claim 10

Algorithm 11 (S′)

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$
Do $|x|$ times:

1. Choose $\pi \leftarrow \Pi_m$ and sends $E = \pi(E_0)$ to $V^*(x)$.
2. Let b be V^*'s answer.
   w.p. $\frac{1}{2}$, find π' such that $E = \pi'(E_b)$, send it to V^* and output V^*'s output.
   Otherwise, rewind the simulation to its first step.

Abort

Claim 12

$S(x) \equiv S'(x)$ for any $x \in GI$.

Proof: ?

Proving Claim 10 cont.

Algorithm 13 (S'')

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$

1. Choose $\pi \leftarrow \Pi_m$ and sends $E = \pi(E_0)$ to $\mathsf{V}^*(x)$.
2. find π' such that $E = \pi'(E_b)$, send it to V^* and output V^*'s output.

Proving Claim 10 cont.

Algorithm 13 (S'')

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$

1. Choose $\pi \leftarrow \Pi_m$ and sends $E = \pi(E_0)$ to $\mathsf{V}^*(x)$.
2. find π' such that $E = \pi'(E_b)$, send it to V^* and output V^*'s output.

Claim 14

$\forall x \in \mathrm{GI}$ it holds that

1. $\mathrm{SD}(\mathsf{S}''(x), \mathsf{S}'(x)) \leq 2^{-|x|}$.

Proving Claim 10 cont.

Algorithm 13 (S'')

Input: $x = (G_0 = ([m], E_0), G_1 = ([m], E_1))$

1. Choose $\pi \leftarrow \Pi_m$ and sends $E = \pi(E_0)$ to $\mathsf{V}^*(x)$.
2. find π' such that $E = \pi'(E_b)$, send it to V^* and output V^*'s output.

Claim 14

$\forall x \in \mathrm{GI}$ it holds that

1. $\mathrm{SD}(\mathsf{S}''(x), \mathsf{S}'(x)) \leq 2^{-|x|}$.
2. $\langle(\mathsf{P}, \mathsf{V}^*(x))\rangle \equiv \mathsf{S}''(x)$.

Proof: ?

Remarks

1. We proved that GI $\in \mathcal{SZKP}$

Remarks

1. We proved that GI $\in \mathcal{SZKP}$
2. Aborting verifiers

Remarks

1. We proved that GI $\in \mathcal{SZKP}$
2. Aborting verifiers
3. Randomized verifiers

Remarks

1. We proved that GI $\in \mathcal{SZKP}$
2. Aborting verifiers
3. Randomized verifiers
4. Negligible soundness error?

Remarks

1. We proved that GI $\in \mathcal{SZKP}$
2. Aborting verifiers
3. Randomized verifiers
4. Negligible soundness error?
5. Auxiliary inputs (for both parties)

Remarks

1. We proved that GI $\in \mathcal{SZKP}$
2. Aborting verifiers
3. Randomized verifiers
4. Negligible soundness error?
5. Auxiliary inputs (for both parties)
6. Perfect $\mathcal{ZK}$ for “expected time simulators"

Remarks

1. We proved that GI $\in \mathcal{SZKP}$
2. Aborting verifiers
3. Randomized verifiers
4. Negligible soundness error?
5. Auxiliary inputs (for both parties)
6. Perfect $\mathcal{ZK}$ for “expected time simulators"
7. “Black box" simulation

Section 3

Black-box ZK

Black-box simulators

Definition 15 (Black-box simulator)

(P, V) is CZKP with black-box simulation for $\mathcal{L}$, if $\exists$ oracle-aided PPT S s.t. for every deterministic polynomial-time V^*,

$$\{(\mathsf{P}(y_x), \mathsf{V}^*(z))(x)\}_{x\in\mathcal{L}} \approx_c \{\mathsf{S}^{\mathsf{V}^*(x,z_x,\cdot)}(x, |z_x|)\}_{x\in\mathcal{L}}$$

where $y_x \in R_{\mathcal{L}}(x)$, $z_x \in \{0,1\}^*$ and $\mathsf{V}^*(x, z_x, \cdot)$ is the next message function of V^* on input x and auxiliary input z_x.

Black-box simulators

Definition 15 (Black-box simulator)

(P, V) is CZKP with black-box simulation for $\mathcal{L}$, if $\exists$ oracle-aided PPT S s.t. for every deterministic polynomial-time V^*,

$$\{(\mathsf{P}(y_x), \mathsf{V}^*(z))(x)\}_{x \in \mathcal{L}} \approx_c \{\mathsf{S}^{\mathsf{V}^*(x, z_x, \cdot)}(x, |z_x|)\}_{x \in \mathcal{L}}$$

where $y_x \in R_{\mathcal{L}}(x)$, $z_x \in \{0,1\}^*$ and $\mathsf{V}^*(x, z_x, \cdot)$ is the next message function of V^* on input x and auxiliary input z_x.
Prefect and statistical variants are defined analogously.

Black-box simulators

Definition 15 (Black-box simulator)

(P, V) is CZKP with black-box simulation for $\mathcal{L}$, if $\exists$ oracle-aided PPT S s.t. for every deterministic polynomial-time V^*,

$$\{(P(y_x), V^*(z))(x)\}_{x\in\mathcal{L}} \approx_c \{S^{V^*(x,z_x,\cdot)}(x, |z_x|)\}_{x\in\mathcal{L}}$$

where $y_x \in R_{\mathcal{L}}(x)$, $z_x \in \{0,1\}^*$ and $V^*(x, z_x, \cdot)$ is the next message function of V^* on input x and auxiliary input z_x. Prefect and statistical variants are defined analogously.

1. “Most simulators" are black box (including the one we gave above)

Black-box simulators

Definition 15 (Black-box simulator)

(P, V) is $\mathcal{CZKP}$ with black-box simulation for $\mathcal{L}$, if $\exists$ oracle-aided PPT S s.t. for every deterministic polynomial-time V^*,

$$\{(\mathsf{P}(y_x), \mathsf{V}^*(z))(x)\}_{x \in \mathcal{L}} \approx_c \{\mathsf{S}^{\mathsf{V}^*(x, z_x, \cdot)}(x, |z_x|)\}_{x \in \mathcal{L}}$$

where $y_x \in R_{\mathcal{L}}(x)$, $z_x \in \{0,1\}^*$ and $\mathsf{V}^*(x, z_x, \cdot)$ is the next message function of V^* on input x and auxiliary input z_x. Prefect and statistical variants are defined analogously.

1. “Most simulators" are black box (including the one we gave above)
2. Strictly weaker then general simulation!